Package 'augSIMEX'

October 12, 2022

Type Package

Title Analysis of Data with Mixed Measurement Error and
Misclassification in Covariates

Version 3.7.4

Date 2020-04-23

Author Qihuang Zhang <qihuang.zhang@uwaterloo.ca>, Grace Y. Yi <yyi@uwaterloo.ca>

Maintainer Qihuang Zhang <qihuang.zhang@uwaterloo.ca>

Description Implementation of the augmented
Simulation-Extrapolation (SIMEX) algorithm pro-
posed by Yi et al. (2015) <doi:10.1080/01621459.2014.922777>
for analyzing the data with mixed measurement error and misclassification. The main
function provides a similar summary output as that of glm() function. Both parametric and
empirical SIMEX are considered in the package.

License GPL (>= 2)

Imports Rcpp (>= 0.12.11), MASS, stats, graphics, Formula, nleqslv

LinkingTo Rcpp

NeedsCompilation yes

Repository CRAN

Date/Publication 2020-04-23 13:50:02 UTC

R topics documented:

---

augSIMEX-package   *R package for Analysis of Data with Mixed Measurement Error and
                           Misclassification in Covariates*

Description

Implementation of the augmented SIMEX algorithm for data with mixed measurement error and
misclassification in covariates. The package is allowing for both parametric SIMEX and empirical
SIMEX.

Details

The SIMEX package implements the method developed by Yi et al. (2015) to adjust for both measure-
ment error and misclassification in the generalized linear models. The main function is augSIMEX
which returns an "augSIMEX" object. The user can summarize the returned augSIMEX object in
a similar format as in glm. The extrapolation can also be visualized through plot function. Other
implementation tools are also provided in the package.

Author(s)

Qihuang Zhang and Grace Y. Yi.

Maintainer: Qihuang Zhang <qihuang.zhang@uwaterloo.ca>

References

Yi G Y, Ma Y, Spiegelman D, et al. Functional and structural methods with mixed measurement
error and misclassification in covariates[J]. Journal of the American Statistical Association, 2015,
110(510): 681-696.

See Also

augSIMEX

---

augSIMEX *Analysis of Data with Mixed Measurement Error and Misclassification in Covariates*

Description

Implementation of the SIMEX algorithm for data with mixed measurement error and misclassification in covariates.

Usage

```
augSIMEX(mainformula = formula(data), mismodel = pi | qi ~ 1,
                 meformula = NULL, family = gaussian, data,
                 validationdata, err.var, mis.var, err.true, mis.true,
                 err.mat = NULL, cppmethod = TRUE, repeated = FALSE,
                 repind = list(), subset, offset, weights, na.action,
                 scorefunction = NULL, lambda = NULL, M = 5, B = 20,
              nBoot = 50, extrapolation = c("quadratic", "linear"), bound = 8,
                 initial = NULL, ...)
```

Arguments

mainformula	an object of class "formula": an object of class "formula": a symbolic description of the model of the response variable, error-prone covariates, and other covariates.
mismodel	an object of class "Formula". A symbolic description in modeling the misclassification rates. See details for the specification of the model.
meformula	an object of "Formula" specifying the measurement error model for each error-prone covariate. The number of responses should equal the number of error-prone covariates. The default choice will be classic additive models. See details for the specification of the model.
family	an object of class "family" (same as in [glm](#) function in stats package).
data	a data frame or a matrix of main data. The variable in the main data includes the response, observed covariates matrix which is subject to measurement error, observed binary covariate vector that is prone to misclassification and may also contain the precisely measured covariates matrix. The default choice includes all covariates that mentioned in the mainformula.
validationdata	a data frame or matrix of validation data. The variable in the model includes an observed covariates matrix of that is subject to measurement error and their corresponding precisely measured covariates, an observed binary covariate vector that is prone to misclassification and their corresponding precisely measured covariates, a covariates matrix of precisely measured covariates.
err.var	a vector of character specifying the name of covariates that are subject to measurement error.

mis.var	a string specifying the variable name of the binary variable subject to misclassification.
err.true	a vector of character specifying the names of the all the counterparts of `err.var` that are precisely measured covariates.
mis.true	a string specifying the variable name of the precisely measured binary variable.
err.mat	a matrix indicating the variance-covariance matrix of generated Normal distribution in simulation step.
cppmethod	a logical value indicating whether solving the score function via C++ functions. The function involves `Rcpp` package. The C++ based method is much faster and more computationally efficient than the R based method. See the discussion section.
repeated	a logic value indicating whether repeated measurements are involved.
repind	a list of vectors of repeated measurement for error-prone covariates. See detail below.
subset	a logical vector indicating which subjects should be included in the fitting.
offset	a numeric vector indicating the offset. The default choice is null.
weights	an optional numeric vector of the weights should be used in the fitting.
na.action	a function indicating what method should be applied to deal with missing data.
scorefunction	a function of score function. To allow for the generality, the users can specify their own score function. It should be a function of parameters, response, covariates, weights and offset. A matrix of score value for each individual and each parameter should be returned in the function. An example is shown in glmscore
lambda	if the `M` value is not specified, the user can specify a positive sequence of lambda directly. The first element is usually set to be 0.
M	the number of predetermined lambda vector if `lambda` is missing. The default is chosen to be 50.
B	the number of the dataset generated in the simulation step. The default value is set to be 200.
nBoot	the number of the iterations repeated in bootstrap. The default value is set to be 100.
extrapolation	specifies the regression model that involves in the extrapolation step. The options include "linear" and "quadratic". The default is set to be "quadratic".
bound	a value or vector specififying the bound for the absolute value of the regression coefficients. During the simulation, the parameters out of bound will be filter out before extrapolation step.
initial	the initial value of the parameters.
...	other arguments that pass into the function.

Details

The misclassification models are set in "Formula" format, where the misclassification rates for both classes of the binary variable is set simultaneously. The left-hand side sets the responses of the misclassification model. The response should always set be to pi (indicating pimodel) and qi

(indicating qimodel), separated by "|". On the right hand side of the formula sets the covariates of the misclassification model. If the covariates for both models are different, use "|" to separate them in the same order as the response. See example.

The measurement error models are also set in "Formula" format. The left-hand side sets the responses of the measurement error model, which should be consistent with the specification of `err.var`. Each response is separated by "|". On the right-hand side of the formula, the covariates are set. If the covariates for each response are different, use "|" to separate the specifications of covariates and the order should correspond to that of the responses of the measurement error model. See example.

In the case of repeated measurements, the users can pick one of the measurements into the formula of the main model. If more than one covariates have multiple replicates, the users should name the vector of repeated measurements in the list of `repind` by the corresponding representative measurement in the formula of the main model. See example.

The number of bootstrap repetitions should be at least 2. i.e., `nBoot>2`. Otherwise, an error might occur.

The examples are mainly for illustration purpose. NA's are possible to be generated because the bootstrap simulation parameter is only set as 2. To obtain precise results, simulation parameters are supposed to be set on a larger scale, which also involves more time for computing.

Value

`coefficients`	the coefficient of the main model after correction.
`vcov`	a adjusted variance-covariance matrix estimated by bootstrap.
`fitted.values`	the fitted mean values, obtained by transforming the linear predictors by the inverse of the link function.
`lambda`	a vector of values that involved in extrapolation step.
`coefmatrix`	a matrix of coefficient before doing extrapolation. Each row correspond to a specified lambda. Each column represent a component of coefficient. This output if for the plotting purpose.
`deviance`	minus twice the maximized log-likelihood.
`null.deviance`	the deviance of null model which only includes intercept and offset term. It is comparable to the deviance.
`df.residual`	the degree of freedom.
`df.null`	the degree of freedom of null model.
`aic`	Akaike's Information Criterion, minus twice the maximized log-likelihood plus twice the number of parameters.
`...`	Other components are the arguments that have been used in the function.

Author(s)

Qihuang Zhang and Grace Y. Yi

References

Yi G Y, Ma Y, Spiegelman D, et al. Functional and structural methods with mixed measurement error and misclassification in covariates[J]. Journal of the American Statistical Association, 2015, 110(510): 681-696.

See Also

glm

Examples

```
### Example 1: Univariate Case
data(ToyUni)
example1<-augSIMEX(mainformula = Y ~ Xstar + Zstar + W, family = binomial(link = logit),
  mismodel = pi|qi ~ W,
  meformula = Xstar ~ X + Z + W,
  data = ToyUni$Main,validationdata = ToyUni$Validation, subset = NULL,
  err.var = "Xstar", mis.var = "Zstar", err.true = "X", mis.true = "Z",
  err.mat = NULL,
  lambda = NULL, M = 5, B = 2, nBoot = 2, extrapolation="quadratic")
summary(example1)

## Without adjustment
example1_naive <- glm(formula = Y ~ Xstar + Zstar + W,
family = binomial(link = logit),data = ToyUni$Main)
summary(example1_naive)

## using  accurate data
example1_true <- glm(Y~Xstar+Zstar+W, family = binomial(link=logit),
   data=ToyUni$True)


### Example 2: Multivariate Case
data(ToyMult)
ErrorFormula<-Xstar.X1|Xstar.X2~-1+X.X1|-1+X.X2    ## measurement error model
example2<-augSIMEX(mainformula = Y~Xstar.X1+Xstar.X2+Zstar+W.W1+W.W2,
  mismodel=pi|qi~X.X1+X.X2+W.W1+W.W2, family = binomial,
  data=ToyMult$Main,meformula=ErrorFormula,
  validationdata=ToyMult$Validation, subset=NULL,
  err.var=c("Xstar.X1","Xstar.X2"), mis.var="Zstar", err.true=c("X.X1","X.X2"),
  mis.true="Z", err.mat = NULL,
  lambda=NULL, M=5, B=2, nBoot=2, extrapolation="quadratic")
summary(example2)


### Example 3
data(ToyRepeat)
example3<-augSIMEX(mainformula = Y~Xstar1+Zstar+W, family = binomial(link=logit),
  mismodel = pi|qi ~ W, meformula = Xstar ~ X + Z + W,
  data=ToyRepeat$Main,validationdata=ToyRepeat$Validation,
  subset=NULL, err.var="Xstar1", mis.var="Zstar", err.true="X", mis.true="Z",
  err.mat = NULL, repeated = TRUE,repind=list(Xstar1=c("Xstar1","Xstar2")),
  lambda=NULL, M=5, B=2, nBoot=2, extrapolation="quadratic")
summary(example3)
```

---

coef.augSIMEX   *Extract the coefficients from the fitted augSIMEX object*

Description

This function returns the vector of fitted coefficients from a fitted augSIMEX object

Usage

```
## S3 method for class 'augSIMEX'

## S3 method for class 'augSIMEX'
coef(object, ...)
```

Arguments

object	the "augSIMEX" object gotten from augSIMEX function.
...	other arguments to pass to the function.

Author(s)

Qihuang Zhang and Grace Y. Yi.

See Also

coef

Examples

```
data(ToyUni)
example <- augSIMEX(mainformula = Y ~ Xstar + Zstar + W, family = binomial(link = logit),
  mismodel = pi|qi ~ W,
  meformula = Xstar ~ X + Z + W,
  data = ToyUni$Main,validationdata = ToyUni$Validation, subset = NULL,
  err.var = "Xstar", mis.var = "Zstar", err.true = "X", mis.true = "Z",
  err.mat = NULL,
  lambda = NULL, M = 5, B = 2, nBoot = 2, extrapolation="quadratic")
coef(example)
```

GeneRepeat   *Example data for univariate error-prone covariates in repeated measurements case*

Description

This dataset gives an example data for the illustration of usage of [augSIMEX](#) function. The data set is adapted from the outbred Carworth Farms White mice data (Parker et. al., 2016). The dataset contains main data and validation data. We are interested to study the association between the response genotype of rs223979909 and the locomotor response to methamphetamine injections, which is subject to mismeasurement.

Tibia_5 - Tibia_30: the repeated measurement of tibia length Tibia_M: the error-prone measurement of tibia length Batch_T: the precisely labeled batch effect Batch_O: the observed batch effect Weight: the body weights of the mice

Usage

```
data(GeneRepeat)
```

Format

A list of two data frames. Main data (rs38916331, Totdist5, Totdist10, Totdist15, Totdist20, Totdist25, Totdist30, Weight, Batch_O) and validation data (Batch_T, Batch_O).

References

Parker C C, Gopalakrishnan S, Carbonetto P, et al. Genome-wide association study of behavioral, physiological and gene expression traits in outbred CFW mice[J]. Nature genetics, 2016, 48(8): 919.

GeneUni   *Example of genetic data for univariate error-prone covariates*

Description

This data set gives an example data for the illustration of usage of [augSIMEX](#) function. The data set is adapted from the outbred Carworth Farms White mice data (Parker et. al., 2016). The dataset contains main data and validation data. We are interested to study the association between the response genotype of rs38916331 and the tibia length, which is subject to mismeasurement.

Tibia_T: the precise measurement of tibia length Tibia_M: the error-prone measurement of tibia length Batch_T: the precisely labeled batch effect Batch_O: the observed batch effect Weight: the body weights of the mice

Usage

```
data(GeneUni)
```

Format

A list of two data frames. Main data (rs38916331, Weight, Tibia_M, Batch_O) and validation data (Tibia_T, Batch_T, Weight, Tibia_M, Batch_O).

References

Parker C C, Gopalakrishnan S, Carbonetto P, et al. Genome-wide association study of behavioral, physiological and gene expression traits in outbred CFW mice[J]. Nature genetics, 2016, 48(8): 919.

---

glmscore   *Score Value in Generalized Linear Model*

Description

This function can be used to calculate the value of score function. This function can serve as a tool to assist the users in debugging their self-defined score function.

Usage

```
glmscore(beta, Y, data, weight, offset, family)
```

Arguments

beta	a vector specifying the value of parameters.
Y	a vector specifying the response.
data	a matrix or data frame of covariates. If this is for score function debugging purpose, see detail for the rearrangement of the dataset.
weight	a vector of weight.
offset	a vector of offset.
family	a glm "family" object.

Details

In a general setting, Y can be treated as a response and data as covariates.

This function helps to debug the user specified score function. The data are needed to be specially arranged such that the score function output can be successfully passed into augSIMEX function in a correct order. The data should be a matrix or data frame of a combination with error-prone covariates (in the order of err.var as in augSIMEX), other covariates and binary variable that is prone to misclassification. The intercept, if considered, should be included the category of "other covariates".

Value

A matrix of n times m, where n is the number of observations and m is the number of parameters. Each entry in the matrix represents the calculated score value for subject i on parameter j. The vector of row sum will be the score value vector.

Author(s)

Qihuang Zhang and Grace Y. Yi.

References

McCullagh P. Generalized linear models[J]. European Journal of Operational Research, 1984, 16(3): 285-292.

See Also

[glm](glm)

Examples

```
### The user specified function to be checked. (logit link in binomial family)
scorefunction=function(beta,Y,data,weight,offset){
  results<-lapply(1:dim(data)[2],
                  FUN=function(i){
                    S<-lapply(1:dim(data)[1],function(x){
                      eta<- matrix(beta,nrow=1)
              return(weight[x]*Y[x]*data[x,i]-weight[x]*exp(eta)/(1+exp(eta))*data[x,i])})
                    return(S)}
  )
  return(matrix(unlist(results),ncol=dim(data)[2]))
}

data(ToyUni)

### Data need to rearranged. See detail.
nsize<-length(ToyUni$Main[,"Y"])
data.in.score<-data.frame(intercept=1,X=ToyUni$Main[,"Xstar"],
                          W=ToyUni$Main[,"W"],Z=ToyUni$Main[,"Zstar"])

## compare. The results should be identical.
glmscore(rep(0,4),ToyUni$Main[,"Y"],data.in.score,rep(1,nsize),
         rep(0,nsize),family=binomial(link=logit))
scorefunction(rep(0,4),ToyUni$Main[,"Y"],data.in.score,rep(1,nsize),rep(0,nsize))
```

---

logLik.augSIMEX   *Extract the likelihood from the fitted augSIMEX object*

Description

This function outputs the likelihood from a fitted augSIMEX object

Usage

```
## S3 method for class 'augSIMEX'

## S3 method for class 'augSIMEX'
logLik(object, ...)
```

Arguments

object	the "augSIMEX" object gotten from augSIMEX function.
...	other arguments to pass to the function.

Author(s)

Qihuang Zhang and Grace Y. Yi.

See Also

logLik

Examples

```
data(ToyUni)
example <- augSIMEX(mainformula = Y ~ Xstar + Zstar + W, family = binomial(link = logit),
  mismodel = pi|qi ~ W,
  meformula = Xstar ~ X + Z + W,
  data = ToyUni$Main,validationdata = ToyUni$Validation, subset = NULL,
  err.var = "Xstar", mis.var = "Zstar", err.true = "X", mis.true = "Z",
  err.mat = NULL,
  lambda = NULL, M = 5, B = 2, nBoot = 2, extrapolation="quadratic")
logLik(example)
```

---

plot.augSIMEX *Plot of Extrapolation*

Description

This function visualizes the extrapolation step. It plots the simulated value for each lambda value and the curve of extrapolation. The function provide the plots for both methods simultaneously, and guides the user to choose a proper extrapolation method.

Usage

```
## S3 method for class 'augSIMEX'

## S3 method for class 'augSIMEX'
plot(x,...)
```

Arguments

x   the "augSIMEX" object gotten from augSIMEX function.

...   other arguments that are passed into the function.

Details

The user may need to adjust the range of y axis for a proper display.

Author(s)

Qihuang Zhang and Grace Y. Yi.

See Also

plot

Examples

```
data(ToyUni)
example<-augSIMEX(mainformula = Y ~ Xstar + Zstar + W, family = binomial(link = logit),
  mismodel = pi|qi ~ W,
  meformula = Xstar ~ X + Z + W,
  data = ToyUni$Main,validationdata = ToyUni$Validation, subset = NULL,
  err.var = "Xstar", mis.var = "Zstar", err.true = "X", mis.true = "Z",
  err.mat = NULL,
  lambda = NULL, M = 5, B = 2, nBoot = 2, extrapolation="quadratic")

plot(example, ylim = c(-1,0.4))
```

---

predict.augSIMEX   *Predict Method for the model fits by augSIMEX*

Description

This function returns the predictions and optionally estimates the standard errors of the predictions from a fitted augSIMEX object.

Usage

```
## S3 method for class 'augSIMEX'

## S3 method for class 'augSIMEX'
predict(object, newdata = NULL, type = c("link", "response","terms"),
se.fit = FALSE, dispersion = NULL, terms = NULL, na.action = na.pass, ...)
```

Arguments

`object`	the "augSIMEX" object gotten from [augSIMEX](#) function.
`newdata`	An optional data frame in which to look for variables with which to predict. If not specified, the prediction will be conducted on the original main data.
`type`	the type of prediction needed. The default is on the scale of the linear predictors; the alternative `"response"` is on the scale of the response variable. The setting is similar to the [predict.glm](#). The `"terms"` option returns a matrix including the fitted values of each term in the model formula on the linear predictor scale.
`se.fit`	a logical variable indicating if standard errors are required.
`dispersion`	a numeric variable specifying the dispersion of the fit to be assumed when computing the standard errors.
`terms`	a character vector specifies which terms are to be returned. This is the case for `type = "terms"`.
`na.action`	function determining what should be done with missing values in `newdata`. The default is to predict NA.
`...`	other arguments that are passed into the function.

Details

The specifications of the aruguments are the same as the setting in glm function. The user may refer to the [predict.glm](#).

Author(s)

Qihuang Zhang and Grace Y. Yi.

See Also

[predict,predict.glm](#)

Examples

```
data(ToyUni)
example <- augSIMEX(mainformula = Y ~ Xstar + Zstar + W, family = binomial(link = logit),
  mismodel = pi|qi ~ W,
  meformula = Xstar ~ X + Z + W,
  data = ToyUni$Main,validationdata = ToyUni$Validation, subset = NULL,
  err.var = "Xstar", mis.var = "Zstar", err.true = "X", mis.true = "Z",
  err.mat = NULL,
  lambda = NULL, M = 5, B = 2, nBoot = 2, extrapolation="quadratic")
predict(example)
```

residuals.augSIMEX   *Residuals of the Fits Made by augSIMEX*

Description

This function outputs the residuals from a fitted augSIMEX object.

Usage

```
## S3 method for class 'augSIMEX'

## S3 method for class 'augSIMEX'
residuals(object, type = c("deviance", "pearson", "working",
    "response", "partial"), ...)
```

Arguments

object	the "augSIMEX" object gotten from [augSIMEX](#) function.
type	a character specifying what type of residuals to be returned. The option includes the deviance, Pearson, working, response and partial.
...	other arguments that are passed into the function.

Author(s)

Qihuang Zhang and Grace Y. Yi.

See Also

[residuals,residuals.glm](#)

Examples

```
data(ToyUni)
example <- augSIMEX(mainformula = Y ~ Xstar + Zstar + W, family = binomial(link = logit),
  mismodel = pi|qi ~ W,
  meformula = Xstar ~ X + Z + W,
  data = ToyUni$Main,validationdata = ToyUni$Validation, subset = NULL,
  err.var = "Xstar", mis.var = "Zstar", err.true = "X", mis.true = "Z",
  err.mat = NULL,
  lambda = NULL, M = 5, B = 2, nBoot = 2, extrapolation="quadratic")
residuals(example)
```

---

summary.augSIMEX   *Summarizing the Adjusted Fits of Generalized Linear Model*

Description

This function is a method for augSIMEX object.

Usage

```
## S3 method for class 'augSIMEX'

## S3 method for class 'augSIMEX'
summary(object, dispersion = NULL, ...)
```

Arguments

object	the "augSIMEX" object gotten from augSIMEX function.
dispersion	the dispersion parameter for the family used.
...	other arguments passed to the function.

Details

The function provides summary output similar to glm.

Value

coefficients	the matrix of coefficients, standard errors, z-values and p-values.
dispersion	the component from arguments of the function

Author(s)

Qihuang Zhang and Grace Y. Yi.

See Also

summary.glm,augSIMEX

Examples

```
## Please see the example in augSIMEX function
```

ToyMult   *Toy example data for multivariate error-prone covariates*

Description

This dataset gives an example data for the illustration of usage of augSIMEX function. The dataset contains main data and validation data. The main data contains response variable Y and error-prone covariates. Validation data have both correctly observed data and error-prone covariates. See example 2 of augSIMEX for the usage.

Usage

```
data(ToyMult)
```

Format

A list of two data frames. Main data, a data frame 1000 observations of 6 variables. Validation data, 500 observations of 8 variables.

ToyRepeat   *Toy example data for univariate error-prone covariates in repeated measurements case*

Description

This data set gives an example data for the illustration of usage of augSIMEX function in adjusting for the case of repeated measurements error. The dataset contains main data and validation data. The main data includes response variable Y and error-prone covariates. Validation data have both correctly observed data and error-prone covariates. See example 3 of augSIMEX for the usage.

Usage

```
data(ToyRepeat)
```

Format

A list of two data frames. Main data, a data frame with 1000 observations of 5 variables. Validation data, 500 observations of 4 variables.

ToyUni   *Toy example data for univariate error-prone covariates*

Description

This data set gives an example data for the illustration of usage of [augSIMEX](#) function. The dataset contains main data, validation data, and true data. True data correspond to the main data, but the all covariates are corrected observed. The main data and true data contains response variable Y. Both main data and validation data contains the error-prone covariates. Validation data and true data have correctly observed data.

Usage

```
data(ToyUni)
```

Format

A list of three data frames. Main data, a data frame 1000 observations of 4 variables. Validation data, 500 observations of 5 variables. True data, 1000 observations of 4 variables.

---

vcov.augSIMEX   *Extract the variance-covariance matrix from the fitted augSIMEX object*

Description

This function return the variance-covariance matrix from a fitted augSIMEX object

Usage

```
## S3 method for class 'augSIMEX'

## S3 method for class 'augSIMEX'
vcov(object, ...)
```

Arguments

object	the "augSIMEX" object gotten from [augSIMEX](#) function.
...	other arguments to pass to the function.

Author(s)

Qihuang Zhang and Grace Y. Yi.

See Also

vcov

Examples

```
data(ToyUni)
example <- augSIMEX(mainformula = Y ~ Xstar + Zstar + W, family = binomial(link = logit),
  mismodel = pi|qi ~ W,
  meformula = Xstar ~ X + Z + W,
  data = ToyUni$Main,validationdata = ToyUni$Validation, subset = NULL,
  err.var = "Xstar", mis.var = "Zstar", err.true = "X", mis.true = "Z",
  err.mat = NULL,
  lambda = NULL, M = 5, B = 2, nBoot = 2, extrapolation="quadratic")
vcov(example)
```

Index